Alice Liu, Emily Kim, Lily Liang

Professor Zhao Yunhua

CSC448 Artificial Intelligence

12.20.2023

Final Report

Abstraction

The objective of this project is to refine the accuracy of our spam detection model, ensuring a high level of precision in distinguishing between spam and legitimate emails. Secondly, our focus extends to increasing security, aiming to decrease the vulnerabilities associated with phishing attacks and malicious emails, thereby fostering a safer online environment for all users. In addition to these primary goals, we aspire to materialize a functional prototype using Viola. As a team, we successfully attained an accuracy score of 97% for the random forest model, solidifying its role as a foundational background model for Viola.

Introduction

Addressing the challenge of differentiating between spam and non-spam emails is the main focus of our project's classification task. Even while current algorithms can categorize emails, spam continues to find its way into our inboxes, suggesting that further improvement is necessary. We focused on developing a classification system that is more accurate and reliable, reducing the amount of spam emails that reach an individual's user's inbox. Throughout our project, we employed six distinct models: Naive Bayes (multinomial naive bayes), SVM, Logistic Regression, Random Forest, Gradient Boosting, and Decision Tree. These models played a crucial role in our efforts to identify the most accurate model for accurate email classification.

Pre-processing

To prepare our data for a model, a crucial step involves data cleaning. Initially, we identified 2 datasets: spam.csv and spam_or_not_spam.csv. Here are the specifics of each datasets:

- spam.csv
    - 2 columns, 5k rows
    - Columns: Category, Message
- spam_or_not_spam.csv
    - 2 columns, 3k rows
    - Columns: Email, Label

Both datasets exhibited similar formatting, prompting us to merge them and create a more extensive dataset for our model. During this combination, meticulous attention was paid to ensure uniformity in format. This involved relabeling columns, standardizing values to represent spam indicators as 0 and 1, removing NULL values, and eliminating any duplications. Further insights into this data combining process are documented in our "Combining Data.ipynb". The resulting dataset is characterized by the following details:

- combined_data_clean.csv
    - 2 columns, 8k rows
    - Columns: Label, Email

Following the meticulous combination of our data, we delved into its exploration and prepared it for preprocessing. We identified 515 emails out of 8,029 to be in foreign languages. Contemplating the option of translation, we decided against it due to potential inaccuracies that could disrupt the model's outcomes. Consequently, we collectively opted to omit these rows from our dataset.

To gain a better understanding, we sought to discern the words or phrases indicative of spam and non-spam classifications. Our approach involved conducting word and frequency counts, employing unary, bigram, and trigram n-gram models on emails categorized as spam and not-spam. To standardize the process, we converted all words to lowercase, tokenized the strings, and eliminated stopwords. Further details on the cleaning and exploration stages can be found in each team member's respective Exploratory Data Analysis (EDA) notebook: Alice EDA, Emily EDA, Lily EDA.

The removal of foreign language data and data preprocessing, encompassing tasks such as removing foreign language datapoints, conversion to lowercase, string tokenization, and stopword removal, was executed in the "PREPROCESSING.ipynb" notebook. The outcomes were then distilled into a new CSV file:

- preprocessed_english.csv
    - 3 columns, 8k rows
    - Columns: Label, Email, processed_email

This file is used in our modeling portion of the project.

Modeling

In our project, we employed a range of machine learning algorithms chosen based on their perceived suitability and potential efficacy for addressing the project requirements.

1. Naive Bayes (Multinomial Naive Bayes)
    a. The Naive Bayes classifier, specifically the Multinomial Naive Bayes variant, was selected for its simplicity and effectiveness in text classification tasks. It is a classification algorithm that's particularly well-suited for text-based data, such as classifying documents or emails into categories like spam or not spam, topics,

sentiment analysis, etc. By leveraging the probabilities of word occurrences, this model assumes independence between features and performs remarkably well even with relatively small datasets. Its computational efficiency and ability to handle large feature spaces made it a suitable choice for our project.

2. Support Vector Machine (SVM)

    a. SVM is a powerful classifier that was utilized due to its ability to effectively separate data points using a hyperplane, optimizing the margin between classes. It excels in high-dimensional spaces and was expected to perform well in our text-based classification task by identifying complex decision boundaries.

3. Logistic Regression

    a. Logistic Regression, aka Logit, is a classification machine learning algorithm that uses labeled data to predict a discrete outcome by assigning a predicted probability to each decision. Its simplicity and interpretability make it a popular choice for binary classification tasks. In this project, it was used as a baseline model to benchmark the performance of more complex algorithms.

4. Random Forest

    a. Random Forest algorithm is an ensemble method based on decision trees, and was employed for its capability to handle non-linear relationships within data and reduce overfitting. By constructing multiple decision trees and aggregating their outputs, this model aimed to enhance classification accuracy and robustness.

5. Gradient Boosting Models

    a. Gradient Boosting Models, including Gradient Boosting Classifier, were chosen for their ability to combine multiple weak learners (typically decision trees) into a strong learner. The iterative nature of boosting techniques helps to correct errors made by previous models, potentially leading to superior predictive performance.

6. Decision Tree
    a. Decision trees classify the examples by sorting them down the tree from the root to some leaf/terminal node, with the leaf/terminal node providing the classification of the example. Decision Trees were explored to create a clear visualization of decision-making processes. These models are intuitive, representing decisions as branches and nodes, making them easy to interpret and explain.

The selection of these diverse models aimed to explore various approaches to classifying spam and non-spam emails. This strategy allowed for a comprehensive evaluation of performance across different algorithmic paradigms, enabling us to find the most suitable model for our specific project. Detailed implementation of the modeling process can be found in "Model Building.ipynb" notebook.

Analysis

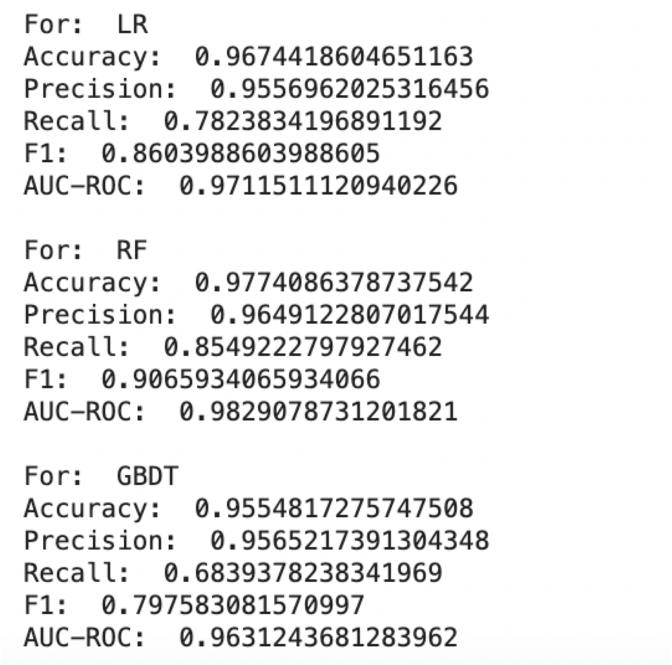
We found that the model that does the best is Random Forest due to our evaluation metrics for classifiers. Our metrics include accuracy, precision, recall, F1 score, and AUC-ROC.

- *Accuracy* is the count of all the predictions we got correct divided by the total number of predictions so the percent of predictions we got correct.
    - A higher accuracy indicates better overall performance.
- *Precision* measures the accuracy of positive predictions made by a model so what proportion of positive identifications was actually correct? Out of all the times our model says "YES", what percentage was correct.
    - A higher precision means fewer false positives
- *Recall* measures the ability of a classifier or model to identify all relevant instances, specifically positive instances so what proportion of actual positives was identified

correctly? Out of all the times THE ACTUAL was "YES", what percentage did you correctly label.

  - A high recall means fewer false negatives

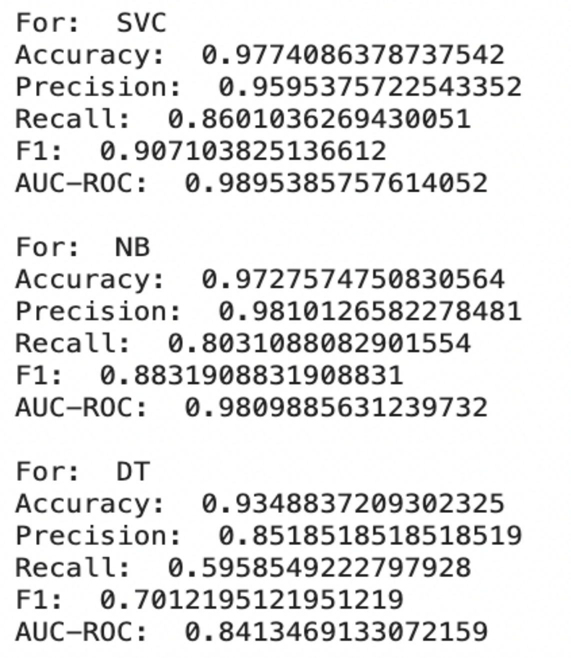
- *F1 Score*: The 'harmonic mean' of precision and recall and can be interpreted as a weighted average of the precision and recall, where an F1 score reaches its best value at 1 and worst score at 0.
  - A high f1 score indicates better trade-off between precision and recall
- *AUC-ROC*: This represents the area under the receiver operating characteristic curve, which measures the model's ability to distinguish between positive and negative instances.
  - A higher ROC-AUC indicates better discrimination between classes.



```
For:  LR
Accuracy:  0.9674418604651163
Precision:  0.9556962025316456
Recall:  0.7823834196891192
F1:  0.8603988603988605
AUC-ROC:  0.9711511120940226

For:  RF
Accuracy:  0.9774086378737542
Precision:  0.9649122807017544
Recall:  0.8549222797927462
F1:  0.9065934065934066
AUC-ROC:  0.9829078731201821

For:  GBDT
Accuracy:  0.9554817275747508
Precision:  0.9565217391304348
Recall:  0.6839378238341969
F1:  0.797583081570997
AUC-ROC:  0.9631243681283962
```



```
For:  SVC
Accuracy:  0.9774086378737542
Precision:  0.9595375722543352
Recall:  0.8601036269430051
F1:  0.907103825136612
AUC-ROC:  0.9895385757614052

For:  NB
Accuracy:  0.9727574750830564
Precision:  0.9810126582278481
Recall:  0.8031088082901554
F1:  0.8831908831908831
AUC-ROC:  0.9809885631239732

For:  DT
Accuracy:  0.9348837209302325
Precision:  0.8518518518518519
Recall:  0.5958549222797928
F1:  0.7012195121951219
AUC-ROC:  0.8413469133072159
```

In our model building, we used a dictionary to store all the models we used and printed the evaluation metrics for each model. There is a screenshot above showing the results where

- NB is the Naive Bayes (multinomial naive bayes) model
- SVC is the SVM model
- LR is the Logistic regression model

- RF is the Random Forest model
- GBDT is the Gradient Boosting Model
- DT is the Decision Tree model

The best model was Random Forest because it got the highest scores for all evaluation metrics.

Random Forest:

- ➢ Accuracy: 97.7%
- ➢ Precision: 96.5%
- ➢ Recall: 85.5%
- ➢ F1 score: 90.7%
- ➢ AUC-ROC: 98.3%

Decision Tree:

- ➢ Accuracy: 93.5%
- ➢ Precision: 85.2%
- ➢ Recall: 60%
- ➢ F1 score: 70.1%
- ➢ AUC-ROC: 84.1%

The Random Forest model performed better than the Decision Tree model, and there were three main reasons why. Compared to a single Decision Tree, Random Forest lowers noise and overfitting by utilizing several trees and consolidating predictions. Unlike a single Decision Tree where it's dependent on certain features. Random Forest improves generalization due to its varied feature subsets and decreased connection between trees. Overall, Random Forest is a more reliable option for better categorization due to its skill at handling complexity and overfitting.

Summary

The goal of our project was to improve the accuracy of our spam classification model. We were able to confirm the Random Forest model’s fundamental function in our prototype by reaching an accuracy score of 97% and a precision of 96% using its model.

- **Datasets and Preprocessing:** We used two different datasets, “spam.csv” and ”spam_or_not_spam.csv”, which required meticulous planning in order to guarantee the consistency of the data to be able to combine the two into one. This involved eliminating duplication, dealing with NULL values, and using text processing methods for model compatibility like vectorization, removing common words, also known as stop words, and tokenization.
- **Model Exploration:** We explored six models – Naive Bayes (multinomial naive bayes), SVM, Logistic Regression, Random Forest, Gradient Boosting, and Decision Tree – each model selected based on how well it could handle particular email classification tasks.
- **Model Performance:** The Random Forest model demonstrated to be the most accurate and best precision compared to the other five models, achieving a 97% accuracy score and a 96% precision score.
- **User Interface Development:** We used the Voilà jupyter server extension to produce a working interface where users can test to see whether their text would be considered as spam or not.

Overall, our results confirmed the importance of using a variety of models to improve the accuracy of email classification. The 97% accuracy rate attained with the Random Forest model highlights its effectiveness and provides a solid basis for further advancement in this modern issue.